

## **REMARKS**

Claims 1, 11, and 12 are pending in the present application. In this Amendment, Applicants cancel claims 2-10 and 13-26 from further consideration in this application. Support for the amendments may be found in originally filed claims 2-10 and the original specification on page 7, line 3, to page 10, line 27. No new matter is added by this Amendment. Applicants do not concede that the subject matter encompassed by the claims prior to this Amendment is not patentable over the art cited by the Examiner. Applicants amend the claims solely to facilitate expeditious prosecution of potentially allowable subject matter. Applicants respectfully reserve the right to pursue the claims as presented prior to this Amendment and additional claims in one or more continuing applications.

Applicants respectfully request reconsideration of the pending claims in view of the following remarks.

### **I. Telephone Interview**

Applicants thank Examiner Rutten for the courtesies extended to Applicants' representative during the telephone interview on July 1, 2008. During the telephone interview, the distinctions of the claims over the cited art were discussed. Examiner Rutten agreed to reconsider the rejection of independent claim 1. The substance of the telephone interview is summarized in the following remarks.

### **II. 35 U.S.C. § 103(a), Alleged Obviousness of Claims 1-9, 12-21, and 24-26**

The Office rejects claims 1-9, 12-21, and 24-26 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Upson et al.* (U.S. Patent No. 5,694,578) in view of *Banning et al.* (U.S. Patent No. 5,421,008). Applicants cancel claims 2-9, 13-21, and 24-26 and amend claim 1 to incorporate features previously presented in claims 2-9. Applicants respectfully traverse this rejection as it applies to current claims 1 and 12.

*Upson* teaches a computer-implemented method and apparatus for converting data according to a selected data transformation. *Upson* states:

The conversion system allows a user to convert data in a given structure into a desired data structure. Initially, the user constructs a visual graphical input template that describes the structure of the input data to the present invention. Only those pieces of data that are necessary for construction of the output structure need be described. Thus, the present invention can be operated in such a manner that uninteresting pieces of an input data structure are ignored.

Thereafter, the user constructs a visual graphical output template which describes the desired structure of the output data. Finally, the user describes the manner in which the output data structure is to be constructed from the input data structure. Such a description is accomplished by selecting pieces of the input data template and "assigning" those pieces to the corresponding pieces of the output template. The present invention will then evaluate the description and generate a data chopping or scribe module 94 which will carry out the data conversion specified by the user. The end result is a data transform.

*Upton*, col. 4, lines 35-53. Thus, *Upton* teaches a data transform that is generated graphically (visual based module). *Upton* also states:

The data transform librarian 88 allows for data transforms to be stored within it. In addition to receiving the p-code fragments 76 and reference table 78 from a given data transform, the data transform librarian 88 is also connected to the visual manager 72 for load/save instances. Each complete data transform contains three pieces: the textual description 86, the p-code fragments 76, and the reference tables 78 of the data transform. A complete data transform will require all three pieces in order for the data scribe module 94 to execute. An incomplete data transform may, however, be stored within the data transform librarian 88. An incomplete data transform may result from work in progress or libraries of clip-templates, which are useful templates or template components that can be utilized.

*Upton*, col. 6, lines 7-20. Thus, *Upton* teaches that all data transforms are stored in the same form. In other words, all data transforms are visual based and stored in the library in a common form.

In contradistinction, the present invention, as recited in claim 1, receives an instruction for selecting the set of transformation modules from a memory and converts each of the set of transformation modules to a common model format. The set of

modules has a first transformation module being a language constructed module and a second transformation module being a visually constructed module. *Upson* does not teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module. Rather, all of the data transforms in *Upson* are visually constructed. See *Upson*, col. 4, lines 35-53. Furthermore, there is no need to convert the data transforms of *Upson* into a common model format, because the data transforms of *Upson* are already stored in a common form.

Furthermore, claim 1 recites that one of the first module or the second module references the other of the first module or the second module. *Upson* does not teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module. See *Upson*, col. 4, lines 35-53. Therefore, *Upson* cannot teach a language constructed transform module referencing a visually constructed transform module or *vice versa*.

The Office Action cites *Banning* as allegedly teaching that text based instructions could be saved in a common data model along with visual based instructions at col. 8, lines 31-35, which states:

One aspect of the invention was the recognition that bidirectional translation or conversion between text based SQL query statements and graphically based visual query representations requires the creation and use of a common data structure.

Thus, *Banning* teaches that translation of text based SQL query statements, in their executable form, and graphically based visual query representations, in their editable form in a graphical user interface, requires a common data structure. Applicants note that the above cited portion of *Banning* does not teach or fairly suggest converting data transform modules of different types to a common model. Rather, *Banning* teaches throughout the disclosure that all SQL statements are created visually. See *Banning*, Abstract, which states:

A method, system and program providing graphical queries and direct manipulation of a database is disclosed. In a preferred form, tables and lists are configured from a database to define a common data structure. Additional, dynamic data structures are employed based on information entered by a user to define various relationships between the dynamic data and the database information. The system

employs a graphical query interface and a relational database to provide an ergonomic, natural interface for a database user.

*Banning* does appear to recognize that translating between the graphical form and its corresponding SQL text requires a common model. See *Banning*, col. 8, lines 31-35. However, *Banning*, like *Upson*, fails to teach or suggest receiving an instruction for selecting the set of transformation modules from a memory and converting each of the set of transformation modules to a common model format.

Furthermore, *Banning*, like *Upson*, fails to teach or suggest receiving a set of transform modules that include a language constructed module and a visually constructed module, because *Banning* teaches only visually constructed SQL statements. See *Banning*, Abstract.

*Upson* and *Banning*, taken individually or in combination, fail to teach or fairly suggest each and every claim feature. Therefore, the proposed combination of *Upson* and *Banning* does not render claim 1, for example, obvious. Because claim 12 depends from claim 1, the same distinctions between *Upson* and *Banning* and claim 1 apply for this claim. In addition, claim 12 recites additional combinations of features not taught or suggested by *Upson* and *Banning*.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1 and 12.

### **III. 35 U.S.C. § 103(a), Alleged Obviousness of Claims 10, 11, 22, and 23**

The Office rejects claims 10, 11, 22, and 23 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Upson* in view of *Banning* and further in view of *Aho et al.* (“Compilers: Principles, Techniques, and Tools”). Applicants cancel claims 10, 22, and 23 and amend claim 1 to incorporate features previously presented in claim 10. Therefore, Applicants respectfully traverse this rejection as it now applies to current claims 1 and 11.

With respect to claim 1, the Office Action acknowledges that *Upson* and *Banning* fail to teach or suggest that the common model format is generic for generation of the executable version for a selected one of a plurality of runtime environments for a data transform engine. The Office Action alleges that *Aho* teaches an intermediate

representation that is used for generation for suitable executables at page 12, which states:

After syntax and semantic analysis, some compilers generate an explicit intermediate representation of the source program. We can think of this intermediate representation as a program for an abstract machine. This intermediate representation should have two important properties: it should be easy to produce, and easy to translate into a target program.

*Aho*, bottom of page 12. While *Aho* does appear to teach an intermediate representation of a source program in the cited portion above, *Aho* does not teach or fairly suggest a common **model** for a **data transform module**.

Furthermore, with respect to claims 1 and 11, *Aho*, like *Upson* and *Banning*, fails to teach or fairly suggest receiving an instruction for selecting the set of transformation modules from a memory and converting each of the set of transformation modules to a common model format, wherein the set of transform modules includes a language constructed module and a visually constructed module. *Upson*, *Banning*, and *Aho* taken individually or in combination, fail to teach or fairly suggest each and every claim feature. Therefore, the proposed combination of *Upson*, *Banning*, and *Aho* does not render claims 1 and 11 obvious.

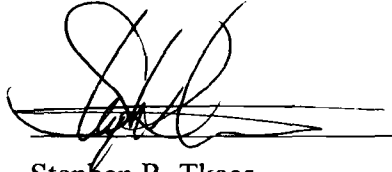
The cited references taken individually teach data transforms, some language based and some visual based. The cited references taken individually teach a common model for translating between language based and visual based transforms. However, the cited references, taken individually **or in combination**, fail to teach or suggest a plurality of data transforms, some language based and some visually based, where a language based transform references a visually based transform or *vice versa*, and converting that plurality of data transforms to a common model format. The cited references, taken alone or in combination, fail to teach or suggest generating an executable version of the converted transforms to form deployment modules and executing the deployment modules in a deployment engine.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1 and 11.

**IV. Conclusion**

It is respectfully urged that the subject application is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'S. Tkacs', is written over a horizontal line.

DATE: August 12, 2008

Stephen R. Tkacs  
Reg. No. 46,430  
**WALDER INTELLECTUAL PROPERTY LAW, P.C.**  
17330 Preston Road, Suite 100B  
Dallas, TX 75252  
(972) 380-9475  
AGENT FOR APPLICANTS